UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/972,321	10/05/2001	Kendell A. Chilton	EMC01-19(01056)	5008

7590   07/30/2004

David E. Huang, Esq.
CHAPIN & HUANG, L.L.C.
Westborough Office Park
1700 West Park Drive
Westborough, MA 01581

EXAMINER
FLEURANTIN, JEAN B

ART UNIT	PAPER NUMBER
2172	

DATE MAILED: 07/30/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>06 May 2004</u>.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1.   This is in response to the amendment filed 6 May 2004, in which claims 1-30 remain

pending for examination. Examiner discusses newly added claims 23-30 in the following

rejection.

Response to Arguments

2.   Applicant's arguments filed 6 May 2004 with respect to claims 1-30 have been fully have

been fully considered but, have been found persuasive only to the extent that the prior art of

record does not specifically teach the limitations "atomically modify the doubly linked list data

structure in accordance with the modify command." However, Kirkman teaches such

limitations.

In response to applicant's argument on page 16, that there is no suggestion to combine the

references, the examiner recognizes that obviousness can only be established by combining or

modifying the teachings of the prior art to produce the claimed invention where there is some

teaching, suggestion, or motivation to do so found either in the references themselves or in the

knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071,

5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir.

1992). In this case, It would have been obvious to one ordinary skill in the art at the time the

invention was made to modify Mason with Hartung to error check. Such modification would

allow the teachings of Mason and Hartung to improve the reliability of the methods and

apparatus for accessing a doubly linked list in a data storage system, and to provide fast access to data, (see Hartung page 1, lines 10-11).

3.   Applicant has amended the Specification to cure minor informality. Therefore, the Objection of the abstract is now withdrawn.

Claim Rejections - 35 USC § 103

4.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-12, 14-19 and 21-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 5,884,098 issued to Mason, Jr. ("hereinafter Mason") in view of U.S. Patent No. 6,581,063 issued to Kirkman ("hereinafter Kirkman").

As per claim 1, Mason discloses "a memory board for a data storage system" (see col. 3, lines 44-46), comprising:

"an interface which is configured to couple to a bus of the data storage system" as a host I/O port configured for connecting to a host computer and a SCSI port interface configures to attach a plurality of disk, (see col. 3, lines 33-40; col. 5, lines 20-21), and column 7, lines 7-9;

"memory which is configured to store a doubly linked list data structure" as a LRU cache memory configures to store doubly linked list data structure, (col. 6, lines 16-18); and

"a memory board control circuit, coupled to the interface and the memory" as a SCSI port

to which a plurality of disk are attached, (see col. 5, lines 20-22),

"the memory board control circuit being configured" as a disk drive in which is

configured for connection to a host computer, (see col. 3, lines 33-34) to:

"receive a modify command from a processor of a data storage system through the

interface" as read and write modify operation, that receive the new blocks of writing data that

may occur in parallel with retrieving through the disk, (see col. 9, lines 40-63), and column 2,

lines 11-16, "the processor being configured to move data within the data storage system" as a

means of moving the new data to the list of valid old data blocks while the obsolete data for the

written blocks are removed from the list of valid old data block, (see col. 9, lines 52-54), and

column 6, lines 24-27). Mason does not explicitly disclose provide a result to the processor of

the data storage system through the interface in response to modifying the doubly linked list data

structure. However, Mason discloses the use of returning the data to the host computer when all

of the requested data has been successfully retrieved into cache memory, (see col. 6, lines 18-29;

col. 8, lines 24-28). It would have been obvious to one having ordinary skill in the art at the time

the invention was made to modify Mason's system, wherein the RAID disk provided therein (see

Mason figure 6, element 407) would incorporate the use of providing the result to the process of

the data storage system, because such modification would have allowed Mason's system the

enhanced capability the methods and apparatus for accessing a doubly linked list in a storage

system, and to provide improvement in disk array controllers systems, (see col. 1, lines 10-11).

While, Mason substantially discloses the claimed subject matter except the claimed atomically

modify the doubly linked list data structure in accordance with the modify command. However,

Kirkman discloses a means of maintaining a boubly-linked list of client object, but with some,

i.e. unique characteristics clients may safely traverse through the list while links are being added

to it and/or removed from it; and multiple clients may independently and concurrently add or

remove links to/from the list. The Shared List object allows all clients to access the list

simultaneously. It provides blocking of tasks at the link level and only those tasks involved in

the individual link conflicts are involved in the blocking, (see Kirkman, col. 6, lines 6-13), and

see col. 4, line 64 to col. 5, line 32. It would have been obvious to one having ordinary skill in

the art at the time the invention was made to modify the combined teachings of Mason and

Kirkman with atomically modify the doubly linked list data structure in accordance with the

modify command. Such modification would allow the teachings of Mason and Kirkman to

provide the ability to traverse the links of a Share list object and to insert or remove links

into/from the list, (see Kirkman col. 7, lines 60-62).

As per claim 2, Mason discloses, "wherein the doubly linked list data structure is a

doubly linked list shared data structure" as a doubly linked list holding a pointer to each cache

block, (see col. 6, lines 16-18).

As per claim 3, Mason discloses "wherein the doubly linked list shared data structure

includes multiple entries" as an LRU cache block list which is a data structure configured as a

doubly linked list holding a pointer to each cache block currently allocated in the cache memory,

such that when a block which is already in the queue is used, and the entry for that block in the

list is moved to the head of the list, in which the entry in the list corresponding to the block

which is used at the earliest time eventually moves to the last position in the list (see col. 6, lines

16-23), "wherein the modify command is a remove instruction" as part of the modify portion of

the RAID level five read-modify-write operation, in which the old data blocks now present in

cache memory are XORed with the parity blocks now present in cache memory to remove the

old data from the parity block, (see col. 9, lines 41-44). Mason does not explicitly disclose

wherein the memory board control circuit is configured to atomically modify the doubly linked

list shared data structure by removing an entry from the doubly linked list shared data structure

in response to the remove instruction. However, Mason states when the new data blocks are

written to the physical disks, the new data blocks become, in effect, "old" data, where pointers to

the cached "new" data are moved to the list of valid old data blocks, while the pointers to the

now-obsolete data for the written blocks are removed from the list of valid old data blocks, (see

col. 9, lines 50-55). It would have been obvious to one ordinary skill in the art at the time the

invention was made to modify Mason's system, wherein the RAID level 5, provided therein (see

Mason's fig. 2) would incorporate the use of removing an entry from the doubly linked list

shared data structure. Such modification would provide Mason's system the enhance capability

of the methods and apparatus for accessing a doubly linked list in a data storage system, and to

provide improvement in disk array controllers systems, (see col. 1, lines 10-11).

As per claim 4, Mason discloses "wherein the doubly linked list shared data structure

includes multiple entries" as an LRU cache block list which is a data structure configured as a

doubly linked list holding a pointer to each cache block currently allocated in the cache memory,

and when a block which is already in the queue is used, and the entry for that block in the list is

moved to the head of the list, in which the entry in the list corresponding to the block which is

used at the earliest time eventually moves to the last position in the list (see col. 6, lines 16-23),

"wherein the modify command is an add instruction" as write all new data which has been

received into the front end cache to the data blocks to the physical disks, in which the front end

cache places pointers to these blocks of cache memory on the list of valid blocks (see figure 6,

element 603; col. 10, lines 12-15). Further, in column 9, lines 41-44, Mason discloses the

modify portion of the RAID level five read-modify-write operation, in which the old data blocks

now present in cache memory are XORed with the parity blocks now present in cache memory to

remove the old data from the parity block. Mason does not explicitly disclose wherein the

memory board control circuit is configured the doubly linked list shared data structure by adding

an entry to the doubly linked list shared data structure in response to the add instruction.

However, Mason states when the new data blocks are written to the physical disks, the new data

blocks become, in effect, "old" data, where pointers to the cached "new" data are moved to the

list of valid old data blocks, while the pointers to the now-obsolete data for the written blocks are

removed from the list of valid old data blocks, (see col. 9, lines 50-55). It would have been

obvious to one ordinary skill in the art at the time the invention was made to modify Mason's

system, wherein the RAID level 5, provided therein (see Mason's fig. 2) would incorporate the

use of removing an entry from the doubly linked list shared data structure. Such modification

would provide Mason's system the enhance capability of the methods and apparatus for

accessing a doubly linked list in a data storage system, and to provide improvement in disk array

controllers systems, (see col. 1, lines 10-11).

As per claim 5, Mason discloses "wherein the doubly linked list shared data structure

includes multiple entries" as an LRU cache block list which is a data structure configured as a

doubly linked list holding a pointer to each cache block currently allocated in the cache memory,

and when a block which is already in the queue is used, and the entry for that block in the list is

moved to the head of the list, in which the entry in the list corresponding to the block which is

used at the earliest time eventually moves to the last position in the list (see col. 6, lines 16-23),

"wherein the modify command is a move instruction" as part of the modify portion of the RAID

level five read-modify-write operation, in which the old data blocks now present in cache

memory are XORed with the parity blocks now present in cache memory to remove the old data

from the parity block, (see col. 9, lines 41-44). Mason does not explicitly disclose wherein the

memory board control circuit is configured to atomically modify the doubly linked list shared

data structure by moving an entry from a first position in the doubly linked list shared data

structure to a second position in the doubly linked list shared data structure in response to the

move instruction. However, Mason states when the new data blocks are written to the physical

disks, the new data blocks become, in effect, "old" data, where pointers to the cached "new" data

are moved to the list of valid old data blocks, while the pointers to the now-obsolete data for the

written blocks are removed from the list of valid old data blocks, (see col. 9, lines 50-55). It

would have been obvious to one ordinary skill in the art at the time the invention was made to

modify Mason's system, wherein the RAID level 5, provided therein (see Mason's fig. 2) would

incorporate the use of removing an entry from the doubly linked list shared data structure. Such

modification would provide Mason's system the enhance capability of the methods and

apparatus for accessing a doubly linked list in a data storage system, and to provide improvement

in disk array controllers systems, (see col. 1, lines 10-11).

As per claim 6, Mason discloses, "and wherein the memory board control circuit is

configured to provide, as the result, a series of transaction outputs respectively corresponding to

the series of individual transactions" as a disk array I/O processor configured to access host data

in the cache memory and in communication with the plurality of disk drives, in which the disk

array I/O processor processing host I/O transactions into disk I/O transactions, (see col. 3, lines

44-51). Mason does not explicitly disclose wherein the memory board control circuit is

configured to atomically modify the doubly linked list shared data structure by performing a

series of individual transactions on the doubly linked list shared data structure. However, Mason

states when the new data blocks are written to the physical disks, the new data blocks become, in

effect, "old" data, where pointers to the cached "new" data are moved to the list of valid old data

blocks, while the pointers to the now-obsolete data for the written blocks are removed from the

list of valid old data blocks, (see col. 9, lines 50-55). It would have been obvious to one ordinary

skill in the art at the time the invention was made to modify Mason's system, wherein the RAID

level 5, provided therein (see Mason's fig. 2) would incorporate the use of removing an entry

from the doubly linked list shared data structure. Such modification would provide Mason's

system the enhance capability of the methods and apparatus for accessing a doubly linked list in

a data storage system, and to provide improvement in disk array controllers systems, (see col. 1,

lines 10-11).

As per claim 7, Mason discloses "a storage system" (see col. 3, lines 44-46), comprising:

"a set of storage devices" (see col. 3, lines 35-36);

"a processor which is configured to move data to and from the set of storage devices"

(see col. 3, lines 44-46);

"a bus coupled to the processor" as the processors may communicate through a system

bus, (see col. 7, lines 7-8); and

"a memory board that includes (i), an interface which couples to the bus " as a SCSI port

to which a plurality of disk are attached, (see col. 5, lines 20-22),

(ii) "memory which is configured to store a doubly linked list data structure" as a LRU

cache memory configures to store doubly linked list data structure, (col. 6, lines 16-18); and

(iii) "a memory board control circuit" as a disk drive in which is configured for

connection to a host computer (see col. 3, lines 33-34), "coupled to the interface and the

memory, the memory board control circuit being configured" as a SCSI port to which a plurality

of disk are attached, (see col. 5, lines 20-22) to:

"receive a modify command from a processor of a data storage system through the

interface" as read and write modify operation, that receive the new blocks of writing data that

may occur in parallel with retrieving through the disk, (see col. 9, lines 40-63), and column 2,

lines 11-16, "the processor being configured to move data within the data storage system" as a

means of moving the new data to the list of valid old data blocks while the obsolete data for the

written blocks are removed from the list of valid old data block, (see col. 9, lines 52-54), and

column 6, lines 24-27). Mason does not explicitly disclose provide a result to the processor of

the data storage system through the interface in response to modifying the doubly linked list data

structure. However, Mason discloses the use of returning the data to the host computer when all

of the requested data has been successfully retrieved into cache memory, (see col. 6, lines 18-29;

col. 8, lines 24-28). It would have been obvious to one having ordinary skill in the art at the time

the invention was made to modify Mason's system, wherein the RAID disk provided therein (see

Mason figure 6, element 407) would incorporate the use of providing the result to the process of

the data storage system, because such modification would have allowed Mason's system the

enhanced capability the methods and apparatus for accessing a doubly linked list in a storage

system, and to provide improvement in disk array controllers systems, (see col. 1, lines 10-11).

While, Mason substantially discloses the claimed subject matter except the claimed atomically

modify the doubly linked list data structure in accordance with the modify command. However,

Kirkman discloses a means of maintaining a boubly-linked list of client object, but with some,

i.e. unique characteristics clients may safely traverse through the list while links are being added

to it and/or removed from it; and multiple clients may independently and concurrently add or

remove links to/from the list. The Shared List object allows all clients to access the list

simultaneously. It provides blocking of tasks at the link level and only those tasks involved in

the individual link conflicts are involved in the blocking, (see Kirkman, col. 6, lines 6-13), and

see col. 4, line 64 to col. 5, line 32. It would have been obvious to one having ordinary skill in

the art at the time the invention was made to modify the combined teachings of Mason and

Kirkman with atomically modify the doubly linked list data structure in accordance with the

modify command. Such modification would allow the teachings of Mason and Kirkman to

provide the ability to traverse the links of a Share list object and to insert or remove links

into/from the list, (see Kirkman col. 7, lines 60-62).

As per claim 8, Mason discloses, "wherein the doubly linked list data structure is a

doubly linked list shared data structure" as a doubly linked list holding a pointer to each cache

block, (see col. 6, lines 16-18).

As per claim 9, Mason discloses "wherein the doubly linked list shared data structure

includes multiple entries" as an LRU cache block list which is a data structure configured as a

doubly linked list holding a pointer to each cache block currently allocated in the cache memory,

such that when a block which is already in the queue is used, and the entry for that block in the

list is moved to the head of the list, in which the entry in the list corresponding to the block

which is used at the earliest time eventually moves to the last position in the list (see col. 6, lines

16-23), "wherein the modify command is a remove instruction" as part of the modify portion of

the RAID level five read-modify-write operation, in which the old data blocks now present in

cache memory are XORed with the parity blocks now present in cache memory to remove the

old data from the parity block, (see col. 9, lines 41-44). Mason does not explicitly disclose

wherein the memory board control circuit is configured to atomically modify the doubly linked

list shared data structure by removing an entry from the doubly linked list shared data structure

in response to the remove instruction. However, Mason states when the new data blocks are

written to the physical disks, the new data blocks become, in effect, "old" data, where pointers to

the cached "new" data are moved to the list of valid old data blocks, while the pointers to the

now-obsolete data for the written blocks are removed from the list of valid old data blocks, (see

col. 9, lines 50-55). It would have been obvious to one ordinary skill in the art at the time the

invention was made to modify Mason's system, wherein the RAID level 5, provided therein (see

Mason's fig. 2) would incorporate the use of removing an entry from the doubly linked list

shared data structure. Such modification would provide Mason's system the enhance capability

of the methods and apparatus for accessing a doubly linked list in a data storage system, and to

provide improvement in disk array controllers systems, (see col. 1, lines 10-11).

As per claim 10, Mason discloses "wherein the doubly linked list shared data structure

includes multiple entries" as an LRU cache block list which is a data structure configured as a

doubly linked list holding a pointer to each cache block currently allocated in the cache memory,

and when a block which is already in the queue is used, and the entry for that block in the list is moved to the head of the list, in which the entry in the list corresponding to the block which is used at the earliest time eventually moves to the last position in the list (see col. 6, lines 16-23), "wherein the modify command is an add instruction" as write all new data which has been received into the front end cache to the data blocks to the physical disks, in which the front end cache places pointers to these blocks of cache memory on the list of valid blocks (see figure 6, element 603; col. 10, lines 12-15). Further, in column 9, lines 41-44, Mason discloses the modify portion of the RAID level five read-modify-write operation, in which the old data blocks now present in cache memory are XORed with the parity blocks now present in cache memory to remove the old data from the parity block. Mason does not explicitly disclose wherein the memory board control circuit is configured the doubly linked list shared data structure by adding an entry to the doubly linked list shared data structure in response to the add instruction. However, Mason states when the new data blocks are written to the physical disks, the new data blocks become, in effect, "old" data, where pointers to the cached "new" data are moved to the list of valid old data blocks, while the pointers to the now-obsolete data for the written blocks are removed from the list of valid old data blocks, (see col. 9, lines 50-55). It would have been obvious to one ordinary skill in the art at the time the invention was made to modify Mason's system, wherein the RAID level 5, provided therein (see Mason's fig. 2) would incorporate the use of removing an entry from the doubly linked list shared data structure. Such modification would provide Mason's system the enhance capability of the methods and apparatus for accessing a doubly linked list in a data storage system, and to provide improvement in disk array controllers systems, (see col. 1, lines 10-11).

As per claim 11, Mason discloses "wherein the doubly linked list shared data structure includes multiple entries" as an LRU cache block list which is a data structure configured as a doubly linked list holding a pointer to each cache block currently allocated in the cache memory, and when a block which is already in the queue is used, and the entry for that block in the list is moved to the head of the list, in which the entry in the list corresponding to the block which is used at the earliest time eventually moves to the last position in the list (see col. 6, lines 16-23), "wherein the modify command is a move instruction" as part of the modify portion of the RAID level five read-modify-write operation, in which the old data blocks now present in cache memory are XORed with the parity blocks now present in cache memory to remove the old data from the parity block, (see col. 9, lines 41-44). Mason does not explicitly disclose wherein the memory board control circuit is configured to atomically modify the doubly linked list shared data structure by moving an entry from a first position in the doubly linked list shared data structure to a second position in the doubly linked list shared data structure in response to the move instruction. However, Mason states when the new data blocks are written to the physical disks, the new data blocks become, in effect, "old" data, where pointers to the cached "new" data are moved to the list of valid old data blocks, while the pointers to the now-obsolete data for the written blocks are removed from the list of valid old data blocks, (see col. 9, lines 50-55). It would have been obvious to one ordinary skill in the art at the time the invention was made to modify Mason's system, wherein the RAID level 5, provided therein (see Mason's fig. 2) would incorporate the use of removing an entry from the doubly linked list shared data structure. Such modification would provide Mason's system the enhance capability of the methods and

apparatus for accessing a doubly linked list in a data storage system, and to provide improvement

in disk array controllers systems, (see col. 1, lines 10-11).

As per claim 12, Mason discloses, "and wherein the memory board control circuit is

configured to provide, as the result, a series of transaction outputs respectively corresponding to

the series of individual transactions" as a disk array I/O processor configured to access host data

in the cache memory and in communication with the plurality of disk drives, in which the disk

array I/O processor processing host I/O transactions into disk I/O transactions, (see col. 3, lines

44-51). Mason does not explicitly disclose wherein the memory board control circuit is

configured to atomically modify the doubly linked list shared data structure by performing a

series of individual transactions on the doubly linked list shared data structure. However, Mason

states when the new data blocks are written to the physical disks, the new data blocks become, in

effect, "old" data, where pointers to the cached "new" data are moved to the list of valid old data

blocks, while the pointers to the now-obsolete data for the written blocks are removed from the

list of valid old data blocks, (see col. 9, lines 50-55). It would have been obvious to one ordinary

skill in the art at the time the invention was made to modify Mason's system, wherein the RAID

level 5, provided therein (see Mason's fig. 2) would incorporate the use of removing an entry

from the doubly linked list shared data structure. Such modification would provide Mason's

system the enhance capability of the methods and apparatus for accessing a doubly linked list in

a data storage system, and to provide improvement in disk array controllers systems, (see col. 1,

lines 10-11).

As per claim 14, Mason discloses "in a memory board of a data storage system" (see col. 3, lines 44-46), "a method for accessing a doubly linked list data structure" (col. 6, lines 7-11), the method comprising the steps of:

"receive a modify command from a processor of a data storage system through a bus of the data storage system" as read and write modify operation, that receive the new blocks of writing data that may occur in parallel with retrieving through the disk, (see col. 9, lines 40-63), and column 2, lines 11-16, "the processor being configured to move data within the data storage system" as a means of moving the new data to the list of valid old data blocks while the obsolete data for the written blocks are removed from the list of valid old data block, (see col. 9, lines 52-54), and column 6, lines 24-27). Mason does not explicitly disclose provide a result to the processor of the data storage system through the interface in response to modifying the doubly linked list data structure. However, Mason discloses the use of returning the data to the host computer when all of the requested data has been successfully retrieved into cache memory, (see col. 6, lines 18-29; col. 8, lines 24-28). It would have been obvious to one having ordinary skill in the art at the time the invention was made to modify Mason's system, wherein the RAID disk provided therein (see Mason figure 6, element 407) would incorporate the use of providing the result to the process of the data storage system, because such modification would have allowed Mason's system the enhanced capability the methods and apparatus for accessing a doubly linked list in a storage system, and to provide improvement in disk array controllers systems, (see col. 1, lines 10-11). While, Mason substantially discloses the claimed subject matter except the claimed atomically modify the doubly linked list data structure in accordance with the modify command. However, Kirkman discloses a means of maintaining a boubly-linked list of

client object, but with some, i.e. unique characteristics clients may safely traverse through the list while links are being added to it and/or removed from it; and multiple clients may independently and concurrently add or remove links to/from the list. The Shared List object allows all clients to access the list simultaneously. It provides blocking of tasks at the link level and only those tasks involved in the individual link conflicts are involved in the blocking, (see Kirkman, col. 6, lines 6-13), and see col. 4, line 64 to col. 5, line 32. It would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the combined teachings of Mason and Kirkman with atomically modify the doubly linked list data structure in accordance with the modify command. Such modification would allow the teachings of Mason and Kirkman to provide the ability to traverse the links of a Share list object and to insert or remove links into/from the list, (see Kirkman col. 7, lines 60-62).

As per claim 15, Mason discloses "wherein the step of atomically modifying the doubly linked list shared data structure" as a means of moving the entry for a particular block in a list to the head of the list and deallocating the cache block to the last entry in the list when the cache memory becomes full, (see col. 6, lines 18-29), and col. 9, lines 40-44, Mason discloses as part of the modify portion of the RAID level five read-modify-write operation, in which the old data blocks now present in cache memory are XORed with the parity blocks now present in cache memory to remove the old data from the parity block. Mason does not explicitly indicate wherein the step of updating, as the doubly linked list shared data structure, a doubly linked list shared data structure in an atomic manner. However, Mason indicates when the new data blocks are written to the physical disks, the new data blocks become, in effect, "old" data, where pointers to the cached "new" data are moved to the list of valid old data blocks, while the

pointers to the now-obsolete data for the written blocks are removed from the list of valid old

data blocks, (see col. 9, lines 50-55). It would have been obvious to one ordinary skill in the art

at the time the invention was made to modify the teachings of Mason with wherein the step of

updating, as the doubly linked list shared data structure, a doubly linked list shared data structure

in an atomic manner. Such modification would incorporate the use of updating, as the doubly

linked list shared data structure, a doubly linked list shared data structure in an atomic manner,

and to improve the reliability of the methods and apparatus for accessing a doubly linked list in a

data storage system, and to provide improvement in disk array controllers systems, (see col. 1,

lines 10-11).

As per claim 16, Mason discloses "wherein the doubly linked list shared data structure

includes multiple entries" as an LRU cache block list which is a data structure configured as a

doubly linked list holding a pointer to each cache block currently allocated in the cache memory,

and when a block which is already in the queue is used, and the entry for that block in the list is

moved to the head of the list, in which the entry in the list corresponding to the block which is

used at the earliest time eventually moves to the last position in the list (see col. 6, lines 16-23),

"wherein the modify command is a remove instruction" as part of the modify portion of the

RAID level five read-modify-write operation, in which the old data blocks now present in cache

memory are XORed with the parity blocks now present in cache memory to remove the old data

from the parity block, (see col. 9, lines 41-44). Mason does not explicitly disclose wherein the

step of updating the doubly linked list shared data structure including the step of removing an

entry from the doubly linked list shared data structure in response to the remove instruction.

However, Mason states when the new data blocks are written to the physical disks, the new data

blocks become, in effect, "old" data, where pointers to the cached "new" data are moved to the

list of valid old data blocks, while the pointers to the now-obsolete data for the written blocks are

removed from the list of valid old data blocks, (see col. 9, lines 50-55). It would have been

obvious to one ordinary skill in the art at the time the invention was made to modify Mason's

system, wherein the RAID level 5, provided therein (see Mason's fig. 2) would incorporate the

use of removing an entry from the doubly linked list shared data structure. Such modification

would provide Mason's system the enhance capability of the methods and apparatus for

accessing a doubly linked list in a data storage system, and to provide improvement in disk array

controllers systems, (see col. 1, lines 10-11).

As per claim 17, Mason discloses "wherein the doubly linked list shared data structure

includes multiple entries" as an LRU cache block list which is a data structure configured as a

doubly linked list holding a pointer to each cache block currently allocated in the cache memory,

and when a block which is already in the queue is used, and the entry for that block in the list is

moved to the head of the list, in which the entry in the list corresponding to the block which is

used at the earliest time eventually moves to the last position in the list (see col. 6, lines 16-23),

"wherein the modify command is an add instruction" as write all new data which has been

received into the front end cache to the data blocks to the physical disks, in which the front end

cache places pointers to these blocks of cache memory on the list of valid blocks (see figure 6,

element 603; col. 10, lines 12-15). Further, in column 9, lines 41-44, Mason discloses the

modify portion of the RAID level five read-modify-write operation, in which the old data blocks

now present in cache memory are XORed with the parity blocks now present in cache memory to

remove the old data from the parity block. Mason does not explicitly disclose wherein the step

of updating the doubly linked list shared data structure includes the step of adding an entry to the

doubly linked list shared data structure in response to the add instruction. However, Mason

states when the new data blocks are written to the physical disks, the new data blocks become, in

effect, "old" data, where pointers to the cached "new" data are moved to the list of valid old data

blocks, while the pointers to the now-obsolete data for the written blocks are removed from the

list of valid old data blocks, (see col. 9, lines 50-55). It would have been obvious to one ordinary

skill in the art at the time the invention was made to modify Mason's system, wherein the RAID

level 5, provided therein (see Mason's fig. 2) would incorporate the use adding an entry to the

doubly linked list shared data structure in response to the add instruction. Such modification

would provide Mason's system the enhance capability of the methods and apparatus for

accessing a doubly linked list in a data storage system, and to provide improvement in disk array

controllers systems, (see col. 1, lines 10-11).

As per claim 18, Mason discloses "wherein the doubly linked list shared data structure

includes multiple entries" as an LRU cache block list which is a data structure configured as a

doubly linked list holding a pointer to each cache block currently allocated in the cache memory,

and when a block which is already in the queue is used, and the entry for that block in the list is

moved to the head of the list, in which the entry in the list corresponding to the block which is

used at the earliest time eventually moves to the last position in the list (see col. 6, lines 16-23),

"wherein the modify command is a remove instruction" as part of the modify portion of the

RAID level five read-modify-write operation, in which the old data blocks now present in cache

memory are XORed with the parity blocks now present in cache memory to remove the old data

from the parity block, (see col. 9, lines 41-44). Mason does not explicitly disclose wherein the

step of updating the doubly linked list shared data structure including the step of moving an entry

from a first position in the doubly linked list shared data structure to a second position in the

doubly linked list shared data structure in response to the remove instruction. However, Mason

states when the new data blocks are written to the physical disks, the new data blocks become, in

effect, "old" data, where pointers to the cached "new" data are moved to the list of valid old data

blocks, while the pointers to the now-obsolete data for the written blocks are removed from the

list of valid old data blocks, (see col. 9, lines 50-55). It would have been obvious to one ordinary

skill in the art at the time the invention was made to modify Mason's system, wherein the RAID

level 5, provided therein (see Mason's fig. 2) would incorporate the use of moving an entry from

a first position in the doubly linked list shared data structure to a second position in the doubly

linked list shared data structure in response to the remove instruction. Such modification would

provide Mason's system the enhance capability of the methods and apparatus for accessing a

doubly linked list in a data storage system, and to provide improvement in disk array controllers

systems, (see col. 1, lines 10-11).

As per claim 19, in addition to claim 18, Mason further discloses, "performing a series of

individual transactions on the doubly linked list shared data structure" as an LRU cache block

list may be a data structure configured as a doubly linked list holding a pointer to each cache

block currently allocated in the cache memory, and when a block which is already in the queue is

used, and the entry for that block in the list is moved to the head of the list, in which the entry in

the list corresponding to the block which is used at the earliest time eventually moves to the last

position in the list (see col. 6, lines 16-23), "and wherein the step of providing the result includes

the step of outputting, as the result, a series of transaction outputs respectively corresponding to

the series of individual transactions" as a disk array I/O processor configured to access host data

in the cache memory and in communication with the plurality of disk drives, in which the disk

array I/O processor processing host I/O transactions into disk I/O transactions, (see col. 3, lines

44-51).

As per claim 21, Mason discloses "a memory board circuit" (see col. 3, lines 44-46), "for

accessing a doubly linked list data structure of a data storage system" (col. 6, lines 7-11), "the

memory board control circuit being mountable to a memory board" as to a plurality of disks

forming a disk drive array (see col. 3, lines 35-36), the memory board control circuit comprising:

"an input port that couples to a bus of the data storage system" as a host I/O port

configured for connecting to a host computer and a SCSI port interface configures to attach a

plurality of disk, (see col. 3, lines 33-40; col. 5, lines 20-21), and column 7, lines 7-9;

"an output port that couples to a bus of the data storage system" as a host I/O port

configured for connecting to a host computer and a SCSI port interface configures to attach a

plurality of disk, (see col. 3, lines 33-40; col. 5, lines 20-21), and column 7, lines 7-9; and

"control logic, connected to the input port and to the output port" as a SCSI port to which

a plurality of disk are attached, (see col. 5, lines 20-21), that is configured to:

"receive a modify command from a processor of a data storage system through the input

port" as read and write modify operation, that receive the new blocks of writing data that may

occur in parallel with retrieving through the disk, (see col. 9, lines 40-63), and column 2, lines

11-16, "the processor being configured to move data within the data storage system" as a means

of moving the new data to the list of valid old data blocks while the obsolete data for the written

blocks are removed from the list of valid old data block, (see col. 9, lines 52-54), and column 6,

lines 24-27). Mason does not explicitly disclose provide a result to the processor of the data

storage system through the interface in response to modifying the doubly linked list data

structure. However, Mason discloses the use of returning the data to the host computer when all

of the requested data has been successfully retrieved into cache memory, (see col. 6, lines 18-29;

col. 8, lines 24-28). It would have been obvious to one having ordinary skill in the art at the time

the invention was made to modify Mason's system, wherein the RAID disk provided therein (see

Mason figure 6, element 407) would incorporate the use of providing the result to the process of

the data storage system, because such modification would have allowed Mason's system the

enhanced capability the methods and apparatus for accessing a doubly linked list in a storage

system, and to provide improvement in disk array controllers systems, (see col. 1, lines 10-11).

While, Mason substantially discloses the claimed subject matter except the claimed atomically

modify the doubly linked list data structure in accordance with the modify command. However,

Kirkman discloses a means of maintaining a boubly-linked list of client object, but with some,

i.e. unique characteristics clients may safely traverse through the list while links are being added

to it and/or removed from it; and multiple clients may independently and concurrently add or

remove links to/from the list. The Shared List object allows all clients to access the list

simultaneously. It provides blocking of tasks at the link level and only those tasks involved in

the individual link conflicts are involved in the blocking, (see Kirkman, col. 6, lines 6-13), and

see col. 4, line 64 to col. 5, line 32. It would have been obvious to one having ordinary skill in

the art at the time the invention was made to modify the combined teachings of Mason and

Kirkman with atomically modify the doubly linked list data structure in accordance with the

modify command. Such modification would allow the teachings of Mason and Kirkman to

provide the ability to traverse the links of a Share list object and to insert or remove links

into/from the list, (see Kirkman col. 7, lines 60-62).

As per claim 22, Mason discloses, "wherein the doubly linked list data structure is a

doubly linked list shared data structure" as a doubly linked list holding a pointer to each cache

block, (see col. 6, lines 16-18).

As per claims 23, 25, 27 and 29, in addition to claim 1, Mason further discloses, "change

the doubly linked list data structure in a non-interruptible single operation" as read and write

modify operation, that receive the new blocks of writing data that may occur in parallel with

retrieving through the disk, (see col. 9, lines 40-63), and column 2, lines 11-16.

As per claims 24, 26, 28 and 30, in addition to claim 1, Mason substantially discloses the

claimed subject matter except the claimed prevent execution of any overlapping context to

alleviate having to coordinate modification of the doubly linked list data structure using shared

structure locking overhead operations. However, Kirkman discloses the use of locks is simple

and effective, but has the potential to affect performance. It may take some time to traverse a list,

searching for a particular element. During this time, all other processes are precluded from

modifying the list, and, depending on the type of lock may be precluded from traversing the list

as well, (see Kirkman col. 2, lines 32-38), and col. 2, line 60 to col. 3, lines 8. It would have

been obvious to one having ordinary skill in the art at the time the invention was made to modify

the combined teachings of Mason and Kirkman prevent execution of any overlapping context to

alleviate having to coordinate modification of the doubly linked list data structure using shared

structure locking overhead operations. Such modification would allow the teachings of Mason

and Kirkman to provide the ability to traverse the links of a Share list object and to insert or

remove links into/from the list, (see Kirkman col. 7, lines 60-62).]

5.   Claims 13 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S.

Patent No. 5,884,098 issued to Mason, Jr. ("hereinafter Mason") as applied to claims 1-12, 14-19

and 21-30 and further in view of European Patent (EP) 0114190 A2 issued to Hartung

("hereinafter Hartung").

As per claim 13, Mason discloses, "wherein the series of transaction outputs provided by

the memory board control circuit includes a first transaction output and a second transaction

output" as the host I/O processor is responsible for receiving commands (input) from the host

computer to the RAID array and transferring data and command status responses (output) from

the RAID array back to the host computer, (see col. 2, lines 3-6); "wherein the processor is

configured to (i) generate a pseudo transaction output based on the first transaction output" as a

host I/O processor in communication with the host I/O port and configured to perform I/O

transactions with the host computer through the host I/O pert, (see col. 3, lines 38-40). Mason

does not explicitly disclose compare the pseudo transaction output to the second transaction

output in order to error check operation of the memory board. However, Hartung discloses

parity error checking on the quality of data transfer, (see Hartung page 21, lines 21-36), and page

26, lines 4-5. It would have been obvious to one ordinary skill in the art at the time the invention

was made to modify Mason with Hartung to error check. Such modification would allow the

teachings of Mason and Hartung to improve the reliability of the methods and apparatus for

accessing a doubly linked list in a data storage system, and to provide fast access to data, (see

Hartung page 1, lines 10-11).

As per claim 20, Mason discloses, "wherein the series of transaction outputs provided by

the memory board control circuit includes a first transaction output and a second transaction

output" as the host I/O processor is responsible for receiving commands (input) from the host

computer to the RAID array and transferring data and command status responses (output) from

the RAID array back to the host computer, (see col. 2, lines 3-6); and wherein the method further

comprises the step of:

"generation of a pseudo transaction output based on the first transaction output" as a host

I/O processor in communication with the host I/O port and configured to perform I/O

transactions with the host computer through the host I/O port, (see col. 3, lines 38-40). Mason

does not explicitly disclose "performing and error handling routine in response to an error

message from the processor resulting from" and "a comparison of the pseudo transaction output

to the second transaction output in order to error check operation of the memory board".

However, Hartung discloses "performing and error handling routine in response to an error

message from the processor resulting from" as a means of using unit 10 with access to the data in

error being controlled in accordance with error recovery procedures, (see Hartung page 9, lines

30-32), and "a comparison of the pseudo transaction output to the second transaction output in

order to error check operation of the memory board" as parity error checking on the quality of

data transfer, (see Hartung page 21, lines 21-36). It would have been obvious to one ordinary

skill in the art at the time the invention was made to modify Mason with Hartung to error check.

Such modification would allow the teachings of Mason and Hartung to improve the reliability of

the methods and apparatus for accessing a doubly linked list in a data storage system, and to

provide fast access to data, (see Hartung page 1, lines 10-11).

CONTACT INFORMATION

6.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Jean B Fleurantin whose telephone number is 703-308-6718.

The examiner can normally be reached on 7:30-6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, John B Breene can be reached on 703-305-9790. The fax phone number for the

organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jean Bolte Fleurantin

July 18, 2004

SHAHID ALAM
PRIMARY EXAMINER